(A2)

"traga a cesta com batatas do porão";
"traga a panela do armário";
se "sala é clara" *então* "coloque avental";
"descasque as batatas";
"devolva a cesta ao porão"

O conceito de algoritmo nos permite reconhecer o mesmo padrão de comportamento nos dois eventos. Na descrição algorítmica deste padrão de comportamento aparece um novo elemento estrutural: o *conectivo condicional se ... então*.

A execução ou não da ação "coloque avental" depende, em cada vez que o algoritmo é executado, do resultado (*verdadeiro* ou *falso*) da inspeção se a sala é clara ou não. O conectivo condicional representa a segunda das estruturas de controle: a alternativa.

A possibilidade de um algoritmo corresponder a dois ou mais acontecimentos distintos (através do uso de um ou mais *se ... então*) nos dá uma amostra de seu poderio. Para tornar ainda mais patente a superioridade do algoritmo sobre o relato de um observador, vamos introduzir o *conectivo repetitivo*.

Suponha que um observador bastante meticuloso descreva o processo "descasca as batatas" em termos da ação primitiva "descasca uma batata". Se um dia a dona-de-casa descascar 13 batatas, e no outro 24, teremos dois relatos distintos:

(R3)

"traz a cesta com batatas do porão";
"traz a panela do armário";
13 vezes {
"descasca uma batata";
"descasca uma batata";
...
"descasca uma batata";
}
"devolve a cesta ao porão"

(R4)

"traz a cesta com batatas do porão";
"traz a panela do armário";
24 vezes {
"descasca uma batata";
"descasca uma batata";
...
"descasca uma batata";
}
"devolve a cesta ao porão"

Para a construção de um algoritmo que corresponda aos dois relatos, precisamos reconhecer um mesmo padrão de comportamento nos dois. Vamos então supor que a dona-de-casa é capaz de reconhecer se o número de batatas descascadas é suficiente para o jantar.

Utilizando apenas os conectivos ";" e *se ... então*, precisamos ainda de mais uma suposição para podermos compor um algoritmo geral, que vem a ser a existência de um número máximo de batatas a serem descascadas em qualquer dia. Se este número for 50, podemos montar o algoritmo

(A3)

"traga a cesta com batatas do porão";
"traga a panela do armário";
se "sala é clara" *então* "coloque avental";
50 vezes {
se "número de batatas é insuficiente" *então*
"descasque uma batata";
se "número de batatas é insuficiente" *então*
"descasque uma batata";
...
se "número de batatas é insuficiente" *então*
"descasque uma batata";
}
"devolva a cesta ao porão"

Ora, além dos óbvios inconvenientes encontrados na escrita de tal algoritmo, temos ainda as seguintes objeções:

a) Ao executar o algoritmo, se a dona-de-casa constatar que cinco batatas foram suficientes, de fato ela não precisaria olhar para a panela outras 45 vezes para se certificar.

b) Muitas vezes é difícil dar um limite superior; pode ser que um dia surjam muitos convidados e que as batatas estejam muito pequenas, ocasionando uma falha do algoritmo.

Para fazer frente a todas essas objeções, vamos introduzir agora o conectivo repetitivo *enquanto ... faça*. Com ele, podemos escrever o seguinte algoritmo:

(A4)

"traga a cesta com batatas do porão";
"traga a panela do armário";
se "sala é clara" *então* "coloque avental";
enquanto "número de batatas é insuficiente" *faça*
"descasque uma batata";
"devolva a cesta ao porão"

O acontecimento provocado pela execução de *enquanto* "inspeção" *faça* "ação" corresponde à execução *repetida* do comando *se* "inspeção" *então* "ação" até a primeira vez em que a inspeção der valor falso (caso em que a "ação" não seria executada).

Com isto vemos o extremo poderio do conceito de algoritmo: um texto estático, capaz de cobrir toda uma classe de acontecimentos, no mesmo nível de detalhe do relato de um observador.

Entretanto, nesta mesma relação entre um texto estático e eventos evoluindo no tempo, reside um grande perigo: é facílimo construir um texto, com toda a aparência de um algoritmo, que não seja executável em tempo finito, tal como: *enquanto* "sala é clara" *faça* "descasque uma batata".

Se assumirmos que o ato de descascar batatas não altera a cor da sala, torna-se impossível a execução deste algoritmo em um tempo finito, por maior que seja.